PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 34

Application Number	09/631,726
Filing Date	08/03/2000
First Named Inventor	Balaram Sinharoy
Art Unit	2183
Examiner Name	Aimee J. Li
Attorney Docket Number	AT9-98-535

ENCLOSURES *(Check all that apply)*

- [x] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below):
  Return Postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Winstead Sechrest & Minick P.C.		
Signature			
Printed name	Robert A. Voigt, Jr.		
Date	06/08/2005	Reg. No.	47,159

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Toni Stanley		
Typed or printed name	Toni Stanley	Date	06/08/2005

PTO/SB/17 (11-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 10/01/2004. Patent fees are subject to annual revision.

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) **500.00**

Complete if Known	
Application Number	09/631,726
Filing Date	08/03/2000
First Named Inventor	Balaram Sinharoy
Examiner Name	Aimee J. Li
Art Unit	2183
Attorney Docket No.	AT9-98-535

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order

☑ Deposit Account ☐ None

Deposit Account Number: 09-0447

Deposit Account Name: IBM Corporation

The Director is hereby authorized to: (check all that apply)

☑ Charge fee(s) indicated below

☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17

☑ Credit any overpayments

to the above-identified deposit account.

☐ Other (please identify): _____

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING FEE

Fee Description	Fee ($)	Small Entity Fee ($)	Fee Paid($)
Utility Filing Fee	790	395	
Design Filing Fee	350	175	
Plant Filing Fee	550	275	
Reissue Filing Fee	790	395	
Provisional Filing Fee	160	80	

Subtotal (1) $ _____

FEE CALCULATION (continued)

2. EXTRA CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20	50	25
Each independent claim over 3	200	100
Multiple dependent claims	360	180
For Reissues, each claim over 20 and more than in the original patent	50	25
For Reissues, each independent claim more than in the original patent	200	100

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____	x _____	= _____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____	x _____	= _____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	Fee ($)	Fee Paid ($)
	_____	_____

Subtotal (2) $ _____

3. OTHER FEES

Fee Description	Fee ($)	Small Entity Fee ($)	Fee Paid($)
1-month extension of time	120	60	
2-month extension of time	450	225	
3-month extension of time	1,020	510	
4-month extension of time	1,590	795	
5-month extension of time	2,160	1,080	
Information disclosure stmt. fee	180	180	
37 CFR 1.17(q) processing fee	50	50	
Non-English specification	130	130	
Notice of Appeal	500	250	
Filing a brief in support of appeal	500	250	500
Request for oral hearing	1,000	500	
Other: _____			

Subtotal (3) $ 500

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 47.159	Telephone 512.370.2832
Name (Print/Type)	Robert A. Voigt, Jr.		Date 06/08/2005

AT9-98-535 PATENT

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Balaram Sinharoy	:	Before the Examiner: Aimee J. Li
Serial No.: 09/631,726	:	Group Art Unit: 2183
Filed: August 3, 2000	:	
	:	IBM Corporation
Title: BRANCH PREDICTION CIRCUITS AND METHODS AND SYSTEMS USING THE SAME	:	Intellectual Property Law 11400 Burnet Road Austin, Texas 78758

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines, Inc., which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. §1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on 6-8, 2005.

Toni Stanley
Signature

Toni Stanley
(Printed name of person certifying)

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellant, Appellant's legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-20 are pending in the Application. Claims 1-20 stand rejected. Claims 1-20 are appealed.

IV. STATUS OF AMENDMENTS

The Appellant's response to the Office Action having a mailing date of February 24, 2004, has been considered, but the Examiner indicated that it did not place the application in condition for allowance because Appellant's arguments were deemed unpersuasive.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In one embodiment of the present invention, a method of generating a global history vector comprises the step of determining if a selected group of instructions contains a branch instruction. Specification, page 3, lines 1-3; Specification, page 9, lines 3-25; Specification, page 36, lines 1-9; Figure 9, step 910. The method may further comprise maintaining a current global history vector in a shift register when the selected group does not contain a branch instruction. Specification, page 3, lines 5-6; Specification, page 9, lines 3-25; Specification, page 36, lines 1-9; Figure 9, steps 912-920. The method may further comprise shifting a first value into the shift register to generate a second vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken. Specification, page 3, lines 6-9; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11

– page 22, line 18; Specification, page 36, line 18 – page 37, line 16; Figure 9, steps 928-938. The method may further comprise shifting a second value into the shift register to generate a second vector when the selected group contains a branch instruction and the selected group does not include a branch instruction predicted as a branch taken. Specification, page 3, lines 9-11; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11 – page 22, line 18; Specification, page 36, lines 10-17; Figure 9, steps 924, 926 and 939.

In another embodiment of the present invention, a method of performing branch predictions comprises the step of indexing a branch history table using a first global history vector associated with a first fetch group of instructions during a first fetch cycle to retrieve a first prediction value. Specification, page 3, lines 13-15; Specification, page 8, lines 16-21; Specification, page 35, lines 7-25. The method may further comprise generating a second global history vector associated with a second fetch group of instructions comprising the substep of retaining the first vector when the first fetch group does not contain at least one branch instruction. Specification, page 3, lines 15-18; Specification, page 36, lines 1-9; Figure 9, steps 912-920. The method may further comprise generating a second global history vector associated with a second fetch group of instructions comprising the substep of appending a bit of a first value to the first vector when the first fetch group has at least one branch instruction predicted to be a branch taken. Specification, page 3, lines 18-19; Specification, page 36, line 18 – page 37, line 16; Figure 9, steps 928-938. The method may further comprise appending a bit of a second value to the first vector when the first group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken. Specification, page 3, lines 19-21; Specification, page 36, line 18 – page 37, line 16; Figure 9, steps 928-938. The method may further comprise indexing the branch history table using the second global history vector during a second fetch cycle to retrieve a second branch prediction value. Specification, page 3, lines 21-23; Specification, page 35, line 7 –

page 37, line 16.

In another embodiment of the present invention, branch processing circuitry comprises a shift register for storing a global history vector. Specification, page 3, line 24 – page 4, line 2; Figure 2A, element 60. The branch processing circuitry may further comprise control circuitry for selectively updating a first global history vector stored in the shift register operable to determine if a selected group of instructions contains a branch instruction. Specification, page 4, lines 2-3; Specification, page 9, lines 3-25; Specification, page 36, lines 1-9; Figure 1, element 54; Figure 3A, element 300; Figure 9, step 910. The branch processing circuitry may further comprise control circuitry for selectively updating a first global history vector stored in the shift register operable to maintain the first global history vector in the shift register when the selected group does not contain a branch instruction. Specification, page 4, lines 3-4; Specification, page 9, lines 3-25; Specification, page 36, lines 1-9; Figure 1, element 54; Figure 3A, element 300; Figure 9, steps 912-920. The branch processing circuitry may further comprise control circuitry for selectively updating a first global history vector stored in the shift register operable to shift a first value into the shift register to generate a second vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken. Specification, page 4, lines 4-7; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11 – page 22, line 18; Specification, page 36, line 18 – page 37, line 16; Figure 1, element 54; Figure 3A, element 300; Figure 9, steps 928-938. The branch processing circuitry may further comprise control circuitry for selectively updating a first global history vector stored in the shift register operable to shifting a second value into the shift register to generate a second vector when the selected group contains a branch instruction and does not contain a branch instruction predicted as a branch taken. Specification, page 4, lines 7-8; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11 – page 22, line 18; Specification, page 36, lines 10-17; Figure 1, element 54; Figure 3A, element 300; Figure 9, steps 924, 926 and 939.

In another embodiment of the present invention, a processing system comprises a microprocessor comprising a branch history table for storing branch prediction values. Specification, page 4, lines 9-10; Figure 3A, elements 301, 302. The microprocessor may further comprise a global history shift register for storing a global branch history vector. Specification, page 4, lines 10-11; Figure 2A, element 60. The microprocessor may further comprise logic for generating an index to the branch history table and accessing prediction values stored therein using selected bits of a branch history vector stored in the shift register. Specification, page 4, lines 11-13; Specification, page 8, lines 16-21; Specification, page 35, lines 7-25; Figure 3A, element 300. The microprocessor may further comprise control circuitry for updating a global branch history vector stored in the shift register and operable to retain a current vector stored in the shift register when a selected fetch group does not contain at least one branch instruction. Specification, page 4, lines 13-16; Specification, page 36, lines 1-9; Figure 3A, element 300; Figure 9, steps 912-920. The microprocessor may further comprise control circuitry for updating a global branch history vector stored in the shift register and operable to shift a bit of a first value into the shift register to generate an updated vector when the selected fetch group has at least one branch instruction predicted to be a branch taken. Specification, page 4, lines 16-18; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11 – page 22, line 18; Specification, page 36, line 18 – page 37, line 16; Figure 3A, element 300; Figure 9, steps 928-938. The microprocessor may further comprise control circuitry for updating a global branch history vector stored in the shift register and operable to shift a bit of a second value into the shift register when the selected fetch group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken. Specification, page 4, lines 18-20; Specification, page 10, line 20 – page 11, lines 12; Specification, page 20, line 11 – page 22, line 18; Specification, page 36, lines 10-17; Figure 3A, element 300; Figure 9, steps 924, 926 and 939.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-18 and 20 stand rejected under 35 U.S.C. §102(e) as being anticipated by Zuraski Jr. et al. (U.S. Patent No. 6,502,188) (hereinafter "Zuraski"). Claim 19 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Zuraski in view of Jerry M. Rosenberg's Dictionary of Computers, Information Processing, and Telecommunications Second Edition ©1987 (hereinafter "Rosenberg").

VII. ARGUMENT

A. Claims 1-18 and 20 are not properly rejected under 35 U.S.C. §102(e) as being anticipated by Zuraski.

Claims 1-18 and 20 have been rejected under 35 U.S.C. §102(e) as being anticipated by Zuraski. Paper No. 9, page 2. Appellant respectfully traverses these rejections for at least the reasons stated below and respectfully requests the Examiner to reconsider and withdraw these rejections.

1. Claims 1, 13 and 17 are not anticipated by Zuraski.

For a claim to be anticipated under 35 U.S.C. §102, each and every claim limitation must be found within the cited prior art reference and arranged as required by the claim. M.P.E.P. §2131.

Appellant respectfully asserts that Zuraski does not disclose "shifting a first value into the shift register to generate a second vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken" as recited in claim 1 and similarly in claims 13 and 17. The Examiner cites column 12, lines 38-42; column 13, lines 7-36; Figures 4 and 5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 3. Appellant respectfully traverses and asserts that Zuraski instead discloses that a taken branch is indicated by a binary one and a not taken branch is indicated by a binary zero. Column 13, lines 12-14. Zuraski further discloses that row A of Figure 4 shows the contents prior to dispatch

of a conditional branch. Column 13, lines 17-18. Zuraski further discloses that row B of Figure 4 contains the contents after dispatch of a conditional branch. Column 13, lines 18-19. Zuraski further discloses that in this case, the conditional branch is predicted taken and a one is shifted into the register. Column 13, lines 19-21. Zuraski further discloses that row C of Figure 4 shows the contents after execution of the conditional branch. Column 13, lines 21-22. Hence, Zuraski discloses shifting a "1" in the shift register if the conditional branch is taken and shifting a "0" in the shift register if the conditional branch is not taken. However, Zuraski does not shift a "1" or a "0" in the shift register based on whether a group of instructions contains a branch instruction predicted as taken.

Furthermore, Zuraski instead teaches that the predicted direction of the branch conveyed by the global predictor storage is shifted into the global history shift register by an update logic. Column 12, lines 38-40. Zuraski further discloses that a binary one may represent a taken branch and a binary zero may represent a not taken branch. Column 12, lines 40-42. There is no language in the cited passage that discloses shifting a value into a shift register to generate a vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken.

Thus, Zuraski does not disclose all of the limitations of claims 1, 13 and 17, and thus Zuraski does not anticipate claims 1, 13 and 17. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "shifting a second value into the shift register to generate a second vector when the selected group contains a branch instruction and the selected group does not include a branch instruction predicted as a branch taken" as recited in claim 1 and similarly in claims 13 and 17. The Examiner cites column 12, lines 38-42; column 13, lines 7-36; Figures 4 and 5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 3. Appellant respectfully traverses. As stated above, Zuraski instead discloses that a taken branch is indicated by a binary one and a not taken branch is

indicated by a binary zero. Column 13, lines 12-14. Zuraski further discloses that row A of Figure 4 shows the contents prior to dispatch of a conditional branch. Column 13, lines 17-18. Zuraski further discloses that row B of Figure 4 contains the contents after dispatch of a single conditional branch. Column 13, lines 18-19. Zuraski further discloses that in this case, the conditional branch is predicted taken and a one is shifted into the register. Column 13, lines 19-21. Zuraski further discloses that row C of Figure 4 shows the contents after execution of the conditional branch. Column 13, lines 21-22. Hence, Zuraski discloses shifting a "1" in the shift register if the conditional branch is taken and shifting a "0" in the shift register if the conditional branch is not taken. However, Zuraski does not shift a "1" or a "0" in the shift register based on whether a group of instructions does not include a branch instruction predicted as taken.

Furthermore, Zuraski instead teaches that the predicted direction of the branch conveyed by the global predictor storage is shifted into the global history shift register by an update logic. Column 12, lines 38-40. Zuraski further discloses that a binary one may represent a taken branch and a binary zero may represent a not taken branch. Column 12, lines 40-42. There is no language in the cited passage that discloses shifting a value into a shift register to generate a vector when the selected group contains a branch instruction and the selected group does not include a branch instruction predicted as a branch taken.

Thus, Zuraski does not disclose all of the limitations of claims 1, 13 and 17, and thus Zuraski does not anticipate claims 1, 13 and 17. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "indexing a branch history table using a first global history vector associated with a first fetch group of instructions during a first fetch cycle to retrieve a first prediction value" as recited in claim 6. The Examiner cites column 1, lines 12-32; column 12, lines 25-32 and Figure 3 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page

5. Appellant respectfully traverses and asserts that Zuraski instead discloses that when a global branch is dispatched, a fetch address is conveyed to the local predictor storage, target array and line buffer. Column 12, lines 25-27. Zuraski further discloses that the fetch address is combined with the contents of the global history shift register to form an index which is conveyed to the global predictor storage. Column 12, lines 27-30. Thus, Zuraski discloses forming an index when a single global branch is dispatched. Zuraski does not disclose indexing a branch history table using a global history vector associated with a fetch group of instructions. Thus, Zuraski does not disclose all of the limitations of claim 6 and thus Zuraski does not anticipate claim 6. M.P.E.P. §2131.

Furthermore, in connection with the rejection of the above-cited claim limitation, the Examiner cites further passages (column 12, lines 27-30; column 12, line 57 – column 13, line 6; column 13, lines 8-25 and column 14, lines 6-47) as support for her rejection of the above-cited claim limitation. Paper No. 9, pages 12-13. There is no language in these passages that discloses indexing a branch history table using a global history vector associated with a fetch group of instructions during a first fetch cycle to retrieve a prediction value. Instead, Zuraski discloses that the fetch address is combined with the contents of the global history shift register to form an index which is conveyed to the global predictor storage. Column 12, lines 27-30. Zuraski further discloses that the update logic modifies the global prediction entry to indicate the behavior of the branch. Column 12, lines 61-63. Zuraski further illustrates the contents of the global history shift register before and after a conditional branch is correctly predicted and executed in Figure 4. However, as stated above, there is no language in these passages that discloses indexing a branch history table using a global history vector associated with a fetch group of instructions during a first fetch cycle to retrieve a prediction value. Thus, Zuraski does not disclose all of the limitations of claim 6 and thus Zuraski does not anticipate claim 6. M.P.E.P. §2131.

Further, the Examiner asserts that a fetch group may be interpreted as being the eight values in the 8 bit shift register in each row as illustrated in Figure 4. Paper No. 9, page 12. Appellant respectfully traverses such an interpretation. The Examiner must provide a basis in fact and/or technical reasoning to support such an interpretation. *Ex parte Levy*. 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that the eight values in the 8 bit shift register in each row as illustrated in Figure 4 discloses a fetch group, and that it would be so recognized by persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Since the Examiner has not provided any objective evidence in support of such an assertion, the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 6. M.P.E.P. § 2141.

Further, the Examiner asserts that a fetch group is disclosed in column 14, lines 6-47 of Zuraski. Paper No. 9, page 13. Appellant respectfully traverses. Zuraski instead discloses a group of non-branch instructions and two predicted taken branch instructions in Figure 8. Column 14, lines 23-29. While Zuraski discloses a group of instructions, there is no language in the cited passage that discloses a fetch group of instructions. Neither does the passage disclose indexing a branch history table using a global history vector associated with such a fetch group of instructions. Neither does the passage disclose indexing a branch history table using a global history vector associated with such a fetch group of instructions during a first fetch cycle to retrieve a prediction value.

Appellant further asserts that Zuraski does not disclose "generating a second history vector associated with a second fetch group of instructions" as recited in claim 6. The Examiner cites column 13, line 37 to column 14, line 5 and Figures 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 4. Appellant respectfully traverses and asserts that Zuraski instead discloses an

illustration of a mispredicted first conditional branch with a subsequent second conditional branch. Zuraski further discloses restoring the state of the global history shift register for conditional branch instructions improperly predicted. However, there is no language in the cited passages of generating a history vector associated with a fetch group of instructions. Thus, Zuraski does not disclose all of the limitations of claim 6 and thus Zuraski does not anticipate claim 6. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "appending a bit of a first value to the first vector when the first fetch group has at least one branch instruction predicted to be a branch taken" as recited in claim 6. The Examiner cites column 12, lines 38-42; column 13, lines 7-36; Figures 4 and 5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 4. Appellant respectfully traverses and asserts that Zuraski instead discloses that a taken branch is indicated by a binary one and a not taken branch is indicated by a binary zero. Column 13, lines 12-14. Zuraski further discloses that row A of Figure 4 shows the contents prior to dispatch of a conditional branch. Column 13, lines 17-18. Zuraski further discloses that row B of Figure 4 contains the contents after dispatch of a conditional branch. Column 13, lines 18-19. Zuraski further discloses that in this case, the conditional branch is predicted taken and a one is shifted into the register. Column 13, lines 19-21. Zuraski further discloses that row C of Figure 4 shows the contents after execution of the conditional branch. Column 13, lines 21-22. Hence, Zuraski discloses shifting a "1" in the shift register if the conditional branch is taken and shifting a "0" in the shift register if the conditional branch is not taken. However, Zuraski does not append a bit of a value to a vector when a fetch group has at least one branch instruction predicted to be a branch taken. Thus, Zuraski does not disclose all of the limitations of claim 6, and thus Zuraski does not anticipate claim 6. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner repeats her argument discussed above (page 10 of Appellant's Appeal

Brief) that Zuraski discloses a fetch group. Paper No. 9, page 13. Appellant respectfully traverses for at least the reasons discussed above.

Appellant further asserts that Zuraski does not disclose "appending a bit of a second value to the first vector when the first group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken" as recited in claim 6. The Examiner cites column 12, lines 38-42; column 13, lines 7-36; Figures 4 and 5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 5. Appellant respectfully traverses. As stated above, Zuraski instead discloses that a taken branch is indicated by a binary one and a not taken branch is indicated by a binary zero. Column 13, lines 12-14. Zuraski further discloses that row A of Figure 4 shows the contents prior to dispatch of a conditional branch. Column 13, lines 17-18. Zuraski further discloses that row B of Figure 4 contains the contents after dispatch of a single conditional branch. Column 13, lines 18-19. Zuraski further discloses that in this case, the conditional branch is predicted taken and a one is shifted into the register. Column 13, lines 19-21. Zuraski further discloses that row C of Figure 4 shows the contents after execution of the conditional branch. Column 13, lines 21-22. Hence, Zuraski discloses shifting a "1" in the shift register if the conditional branch is taken and shifting a "0" in the shift register if the conditional branch is not taken. However, Zuraski does not append a bit of a value to a vector when the fetch group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken. Thus, Zuraski does not disclose all of the limitations of claim 6, and thus Zuraski does not anticipate claim 6. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner repeats her argument discussed above (page 10 of Appellant's Appeal Brief) that Zuraski discloses a fetch group. Paper No. 9, page 13. Appellant respectfully traverses for at least the reasons discussed above.

Appellant further asserts that Zuraski does not disclose "retain a current vector in said shift register when a selected fetch group does not contain at least one branch instruction" as recited in claim 17. The Examiner cites column 2, lines 26-27; column 4, lines 36-40; column 5, lines 23-67 and Figure 1 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 9. The Examiner further states that "maintaining a current global history vector is inherent, since the vector is not changed in the shift register unless there is a branch." Paper No. 9, page 9. Appellant respectfully traverses the implied assertion that Zuraski inherently discloses retaining a current vector stored in the shift register when a selected fetch group does not contain at least one branch instruction. As stated above, Zuraski does not disclose selecting a fetch group. Appellant respectfully points out that the Examiner must provide a basis in fact and/or technical reasoning to assert that Zuraski inherently discloses retaining a current vector stored in the shift register when a selected fetch group does not contain at least one branch instruction. *Ex parte Levy*. 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, in order for the Examiner to establish inherency, the Examiner must provide extrinsic evidence that must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Inherency, however, may not be established by probabilities or possibilities. *Id.* The mere fact that a certain thing may resolve from a given set of circumstances is not sufficient. *Id.* Therefore, the Examiner must support her inherency argument with objective evidence meeting the above requirements. Since the Examiner has not provided any objective evidence in support of her inherency argument, the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 17. M.P.E.P. § 2141.

2. Claims 2-5, 7-12, 14-16, 18 and 20 are not anticipated by Zuraski for at least the reasons that claims 1, 6, 13 and 17 are not anticipated by Zuraski.

Claims 2-5 depend from claim 1 and hence are not anticipated by Zuraski for at least the reasons that claim 1 is not anticipated by Zuraski as discussed in Section A(1). Claims 7-12 depend from claim 6 and hence are not anticipated by Zuraski for at least the reasons that claim 6 is not anticipated by Zuraski as discussed in Section A(1). Claims 14-16 depend from claim 13 and hence are not anticipated by Zuraski for at least the reasons that claim 13 is not anticipated by Zuraski as discussed in Section A(1). Claims 18 and 20 depend from claim 17 and hence are not anticipated by Zuraski for at least the reasons that claim 17 is not anticipated by Zuraski as discussed in Section A(1).

3. Claim 2 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "storing the generated value in an entry in a branch instruction queue associated with the selected group of instructions" as recited in claim 2. The Examiner cites to column 10, lines 5-6 and Figure 3 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 3. Appellant respectfully traverses and asserts that Zuraski discloses that the global history shift register is coupled to line buffer 210. However, there is no language in the cited passage that discloses that the value generated based on whether a group of instructions includes or does not include a branch instruction predicted as taken is stored in line buffer 210. Thus, Zuraski does not disclose all of the limitations of claim 2 and thus Zuraski does not anticipate claim 2. M.P.E.P. §2131.

In connection with the rejection of the above-cited claim limitation, the Examiner cites column 10, lines 47-49 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 14. Appellant respectfully traverses. Zuraski instead

discloses that a signal is conveyed to the line buffer which stores a global history. Column 10, lines 47-48. However, there is no language in the cited passage that discloses storing a generated value in an entry in a branch instruction queue. Neither is there any language in the cited passage that discloses storing a generated value in an entry in a branch instruction queue associated with a group of instructions. Thus, Zuraski does not disclose all of the limitations of claim 2 and thus Zuraski does not anticipate claim 2. M.P.E.P. §2131.

4. Claim 3 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "correcting the generated vector upon a misprediction comprising the substeps of: retrieving a selected number of bits of the vector stored from the branch instruction queue into the shift register; and shifting an updated history bit into the shift register" as recited in claim 3. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 3. Appellant respectfully traverses and asserts that Zuraski instead discloses that line buffer 210 entry is used to update or repair the branch prediction and global history upon retirement or misprediction. Column 10, lines 49-51. Zuraski further discloses that in the case of a mispredicted branch, the global history register is updated by modifying the history bit representing the mispredicted branch from a binary one to a zero. Column 13, lines 34-36. Hence, Zuraski does not disclose shifting an updated history bit into the shift register. Instead, Zuraski discloses modifying the history bit to be a different logic value. Further, while Zuraski discloses that line buffer 210 entry is used to update or repair the branch prediction and global history upon retirement or misprediction, the Examiner has not cited to any passage in Zuraski that discloses retrieving a selected number of bits of a vector stored in line buffer 210 into the shift register. Thus, Zuraski does not disclose

all of the limitations of claim 3, and thus Zuraski does not anticipate claim 3. M.P.E.P. §2131.

Furthermore, with respect to the above-cited claim rejection, Zuraski instead discloses restoring the state of the register at the time of prediction of the first conditional branch with the history changed from a binary one to a binary zero to reflect the not taken branch. Column 13, lines 50-53. Zuraski further discloses that the history of the second conditional branch is no longer present in the history register as it represents an erroneous instruction stream. Column 13, lines 53-56. Zuraski further discloses that in this case it was necessary to right-shift the history in the register and restore the history of a branch which was previously left-shifted out. Column 13, lines 56-58. Again, Zuraski discloses modifying the history bit representing the mispredicted branch from a binary one to a zero. Zuraski does not disclose shifting an updated history bit into the shift register but instead discloses right-shifting the history in the register and restoring the history of a branch which was previously left-shifted out. Thus, Zuraski does not disclose all of the limitations of claim 3, and thus Zuraski does not anticipate claim 3. M.P.E.P. §2131.

5. Claim 5 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "wherein the selected group of instructions comprises eight instructions" as recited in claim 5. The Examiner cites column 12, lines 38-42; column 13, lines 7-36 and Figures 4-5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 4. Appellant respectfully traverses and asserts that Zuraski instead discloses an 8 bit global history shift register. However, while the register is an 8 bit register, there is no language in the cited passages that discloses determining if a selected group of instructions contains a branch instruction where the selected group of instructions comprises eight instructions. Hence, there is language in the cited passage that

discloses determining if a branch instruction is contained in a group of eight instructions. Thus, Zuraski does not disclose all of the limitations of claim 5 and thus Zuraski does not anticipate claim 5. M.P.E.P. §2131.

6. Claim 7 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "storing the first and second vectors in an entry of a branch history queue associated with the first fetch group" as recited in claim 7. The Examiner cites to column 10, lines 5-6 and Figure 3 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 5. Appellant respectfully traverses and asserts that Zuraski discloses that the global history shift register is coupled to line buffer 210. However, there is no language in the cited passage that discloses that the vectors generated based on whether a fetch group contains or does not contain a branch instruction predicted to be a branch taken is stored in line buffer 210. Thus, Zuraski does not disclose all of the limitations of claim 7 and thus Zuraski does not anticipate claim 7. M.P.E.P. §2131.

7. Claim 8 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "detecting a branch misprediction based on the first prediction value" as recited in claim 8. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 5. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages of detecting a branch misprediction based on a prediction value that was retrieved from a branch history table. The Examiner had previously asserted that global predictor storage 205 teaches a branch history table. Paper No. 9, page 4. However, there is no language in the cited passages that a prediction value from global predictor storage 205 is used to detect a branch

misprediction. Thus, Zuraski does not disclose all of the limitations of claim 8 and thus Zuraski does not anticipate claim 8. M.P.E.P. §2131.

Further, the Examiner asserts that Zuraski discloses the above-cited claim limitation on pages 14-15 of Paper No. 9. However, the Examiner did not provide any evidence, e.g., passages in Zuraski, to support her assertion. The Examiner must provide evidence that a reference expressly or inherently describes each and every element as set forth in the claim. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "retrieving the first and second vectors from the branch history queue" as recited in claim 8. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 5. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages that discloses retrieving vectors from a branch history queue. Thus, Zuraski does not disclose all of the limitations of claim 8 and thus Zuraski does not anticipate claim 8. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "indexing the branch history table using the first vector to correct the first prediction value" as recited in claim 8. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 5. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages of indexing a branch history table using a vector to correct a prediction value. The Examiner had previously asserted that global predictor storage 205 teaches a branch history table. Paper No. 9 page 4. However, there is no language in the cited passages that discloses indexing in global predictor storage 205 using a vector to correct a prediction value. Thus, Zuraski does

not disclose all of the limitations of claim 8 and thus Zuraski does not anticipate claim 8. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "appending a corrected bit to the second vector to generate a corrected branch history vector" as recited in claim 8. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 6. Appellant respectfully traverses and asserts that Zuraski instead discloses that the global history register is updated by modifying the history bit representing the mispredicted branch from a binary one to a zero in the case of mispredicting a conditional branch to be taken instead of not taken. Zuraski illustrates modifying the history bit from a binary one to a zero in Figure 5. Hence, discloses replacing one value with another value in the case of a misprediction. However, Zuraski does not disclose appending a corrected value to the vector in the global history register. Thus, Zuraski does not disclose all of the limitations of claim 8 and thus Zuraski does not anticipate claim 8. M.P.E.P. §2131.

8. Claim 9 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "wherein said fetch cycle precedes the second fetch cycle by three fetch cycles" as recited in claim 9. The Examiner cites column 1, lines 13-20 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 6. Appellant respectfully traverses and asserts that Zuraski instead discloses that a clock cycle refers to an interval of time accorded to various stages of an instruction processing pipeline within the microprocessor. The language in the cited passage is not specific as to a number of fetch cycles. Neither is there any language in the cited passage disclosing a number of three fetch cycles between a first fetch cycle (used to retrieve a first prediction value) and a second fetch cycle (used to retrieve a second prediction value). Thus, Zuraski does not

disclose all of the limitations of claim 9 and thus Zuraski does not anticipate claim 9. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner asserts that the above-stated phrase "[n]either is there any language in the cited passage disclosing a number of three fetch cycles between a first fetch cycle (used to retrieve a first prediction value) and a second fetch cycle (used to retrieve a second prediction value" made by Appellant is reading limitations into the claim. Paper No. 9, page 15. Appellant respectfully traverses. Claim 9 depends from claim 6 which states "during a second fetch cycle to retrieve a second branch prediction value" and "during a first fetch cycle to retrieve a first prediction value." Hence, Appellant is not reading limitations into the claim.

Further, the Examiner asserts that the teaching of choosing the shortest possible clock cycle consistent with the design (Zuraski, column 1, lines 13-20) may be interpreted as three fetch cycles. Paper No. 9, page 15. Appellant respectfully traverses. The Examiner must provide a basis in fact and/or technical reasoning to support such an interpretation. *Ex parte Levy*. 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that the teaching of choosing the shortest possible clock cycle consistent with the design (Zuraski, column 1, lines 13-20) may be interpreted as three fetch cycles, and that it would be so recognized by persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Since the Examiner has not provided any objective evidence in support of such an assertion, the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 9. M.P.E.P. § 2141.

9. Claim 12 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "wherein said substeps of appending comprise the substeps of shifting a bit into a shift register storing the second vector" as recited in claim 12. The Examiner cites column 12, lines 38-42; column 13, lines 7-36 and Figures 4-5 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 6. Appellant respectfully traverses. As stated above, Zuraski instead discloses that the global history register is updated by modifying the history bit representing the mispredicted branch from a binary one to a zero in the case of mispredicting a conditional branch to be taken instead of not taken. Zuraski illustrates modifying the history bit from a binary one to a zero in Figure 5. Hence, discloses replacing one value with another value in the case of a misprediction. However, Zuraski does not disclose shifting a corrected value to the vector in the global history register. Thus, Zuraski does not disclose all of the limitations of claim 12 and thus Zuraski does not anticipate claim 12. M.P.E.P. §2131.

10. Claim 15 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "a queue for storing said first and said second vectors" as recited in claim 15. The Examiner cites to column 10, lines 5-6 and Figure 3 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 8. Appellant respectfully traverses and asserts that Zuraski discloses that the global history shift register is coupled to line buffer 210. However, there is no language in the cited passage that discloses that the vectors generated based on whether a group of instructions contains or does not contain a branch instruction predicted to be a branch taken is stored in line buffer 210. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "accessing said vectors from said queue" as recited in claim 15. The Examiner cites column 10, lines 5-6 and 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 8. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages that discloses accessing vectors from a queue. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner cites column 12, line 57 – column 13, line 6 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 16. Appellant respectfully traverses. Zuraski instead discloses that line buffer (element 210) conveys the corresponding branch information to the update logic (element 202). There is no language that the update logic accesses the line buffer as asserted by the Examiner. Paper No. 9, page 16. Instead, Zuraski simply teaches that the line buffer conveys branch information to the update logic. There is no language in the cited passage that discloses accessing vectors from a queue. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "indexing said branch history table with said first vector and updating a corresponding entry with a corrected prediction value" as recited in claim 15. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 8. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages of indexing a branch history table using a vector and updating an entry with a corrected a prediction value. The Examiner had previously asserted that global

predictor storage 205 teaches a branch history table. Paper No. 9, page 4. However, there is no language in the cited passages that discloses indexing in global predictor storage 205 using a vector and updating an entry in global predictor storage 205 with a corrected prediction value. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "updating a vector in said shift register with said second vector" as recited in claim 15. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 8. Appellant respectfully traverses and asserts that Zuraski instead discloses that the global history register is updated by modifying the history bit representing the mispredicted branch from a binary one to a zero in the case of mispredicting a conditional branch to be taken instead of not taken. Zuraski illustrates modifying the history bit from a binary one to a zero in Figure 5. Hence, discloses replacing one value with another value in the case of a misprediction. However, Zuraski does not disclose updating a vector in a shift register with another vector that was generated by shifting a value into the shift register when the group of instructions contains a branch instruction and does not contain a branch instruction predicted as a branch taken. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "shifting the corrected prediction value into said shift register" as recited in claim 15. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 8. Appellant respectfully traverses and asserts that Zuraski instead discloses that the global history register is updated by modifying the history bit representing the mispredicted branch from a binary one to a zero in the case of mispredicting a conditional branch to be taken instead of not taken. Zuraski illustrates modifying the

history bit from a binary one to a zero in Figure 5. Hence, discloses replacing one value with another value in the case of a misprediction. However, Zuraski does not disclose shifting a corrected value into the shift register. Thus, Zuraski does not disclose all of the limitations of claim 15 and thus Zuraski does not anticipate claim 15. M.P.E.P. §2131.

11. Claim 18 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "a branch instruction queue having a plurality of entries each associated with a fetch group for storing at least first and second corresponding global history vectors" as recited in claim 18. The Examiner cites column 10, lines 5-6 and Figure 3 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 10. Appellant respectfully traverses and asserts that Zuraski instead discloses a global shift register coupled to global predictor storage 205 and line buffer 210. However, there is no language as to line buffer 210 or global predictor storage 205 (Appellant is unsure as to which unit the Examiner alleges as disclosing a branch instruction queue) having a plurality of entries where each entry is associated with a fetch group. Further, there is no language in the cited passage of line buffer 210 or global predictor storage 205 (Appellant is unsure as to which unit the Examiner alleges as disclosing a branch instruction queue) storing global history vectors. Thus, Zuraski does not disclose all of the limitations of claim 18 and thus Zuraski does not anticipate claim 18. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "detecting a misprediction associated with a said prediction value retrieved from said branch history table and corresponding to said first global history vector in said branch instruction queue" as recited in claim 18. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as

disclosing the above-cited claim limitation. Paper No. 9, page 10. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages of detecting a branch misprediction based on a prediction value that was retrieved from a branch history table. The Examiner had previously asserted that global predictor storage 205 discloses a branch history table. Paper No. 7, page 4. However, there is no language in the cited passages that a prediction value from global predictor storage 205 is used to detect a branch misprediction. Thus, Zuraski does not disclose all of the limitations of claim 18 and thus Zuraski does not anticipate claim 18. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "retrieving said first vector from said branch instruction queue and accessing a corresponding entry in said branch history table to correct said prediction value stored therein" as recited in claim 18. The Examiner cites column 10, lines 49-51; column 13, line 26 – column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 10. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. However, there is no language in the cited passages of retrieving a vector from a branch instruction queue. Neither was there any language in the cited passages of accessing an entry in a branch history table to correct a prediction value stored therein. The Examiner had previously asserted that global predictor storage 205 discloses a branch history table. Paper No. 9, page 4. However, there is no language in the cited passages that an entry in global predictor 205 is accessed to correct a prediction value. Thus, Zuraski does not disclose all of the limitations of claim 18 and thus Zuraski does not anticipate claim 18. M.P.E.P. §2131.

Appellant further asserts that Zuraski does not disclose "retrieving and modifying said second vector to generate a corrected vector in said shift register" as recited in claim 18. The Examiner cites column 10, lines 49-51; column 13, line 26 –

column 14, line 5; and Figures 3, 5, 6 and 7 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 10. Appellant respectfully traverses and asserts that Zuraski instead discloses updating a global history register after a misprediction. There is no language in the cited passages of retrieving and modifying a vector from a branch instruction queue to generate a corrected vector in the shift register. Thus, Zuraski does not disclose all of the limitations of claim 18 and thus Zuraski does not anticipate claim 18. M.P.E.P. §2131.

12. Claim 20 is not anticipated by Zuraski.

Appellant respectfully asserts that Zuraski does not disclose "wherein said fetch group comprises eight instructions" as recited in claim 20. The Examiner cites column 4, lines 12-19 of Zuraski as disclosing the above-cited claim limitation. Paper No. 9, page 10. Appellant respectfully traverses and asserts that Zuraski instead discloses a byte comprises 8 binary bits. This is not the same as a fetch group comprising eight instructions. Thus, Zuraski does not disclose all of the limitations of claim 20 and thus Zuraski does not anticipate claim 20. M.P.E.P. §2131.

B. Claim 19 is not properly rejected under 35 U.S.C. §103(a) as being unpatentable over Zuraski in view of Rosenberg.

The Examiner rejects claim 19 under 35 U.S.C. §103(a) as being unpatentable over Zuraski in view of Rosenberg. Paper No. 9, page 11. Appellant respectfully asserts that claim 19 is allowable as claim 17[1] is allowable for at least the reasons stated above in Section A.

[1] Applicant respectfully notes that claim 19 depends from independent claim 17.

VIII. CONCLUSION

For the reasons noted above, the rejections of claims 1-20 are in error. Appellant respectfully requests reversal of the rejections and allowance of claims 1-20.

Respectfully submitted,
WINSTEAD SECHREST & MINICK P.C.
Attorneys for Appellant

By: _____
Robert A. Voigt, Jr.
Reg. No. 47,159
Kelly K. Kordzik
Reg. No. 36,571

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

APPENDIX

1. A method of generating a global history vector comprising the steps of:

determining if a selected group of instructions contains a branch instruction;

maintaining a current global history vector in a shift register when the selected group does not contain a branch instruction;

shifting a first value into the shift register to generate a second vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken; and

shifting a second value into the shift register to generate a second vector when the selected group contains a branch instruction and the selected group does not include a branch instruction predicted as a branch taken.

2. The method of claim 1 and further comprising the step of storing the generated value in an entry in a branch instruction queue associated with the selected group of instructions.

3. The method of claim 2 and further comprising the step of correcting the generated vector upon a misprediction comprising the substeps of:

retrieving a selected number of bits of the vector stored from the branch instruction queue into the shift register; and

shifting an updated history bit into the shift register.

4. The method of claim 1 wherein the first value comprises a logic 1 and the second value is a logic 0.

5. The method of claim 1 wherein the selected group of instructions comprises eight instructions.

6. A method of performing branch predictions comprising the steps of:

indexing a branch history table using a first global history vector associated with a first fetch group of instructions during a first fetch cycle to retrieve a first prediction value;

generating a second global history vector associated with a second fetch group of instructions comprising the substeps of:

retaining the first vector when the first fetch group does not contain at least one branch instruction;

appending a bit of a first value to the first vector when the first fetch group has at least one branch instruction predicted to be a branch taken;

appending a bit of a second value to the first vector when the first group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken; and

indexing the branch history table using the second global history vector during a second fetch cycle to retrieve a second branch prediction value.

7. The method of claim 6 and further comprising the step of storing the first and second vectors in an entry of a branch history queue associated with the first fetch group.

8. The method of claim 7 and further comprising the steps of:

detecting a branch misprediction based on the first prediction value;

retrieving the first and second vectors from the branch history queue;

indexing the branch history table using the first vector to correct the first prediction value; and

appending a corrected bit to the second vector to generate a corrected branch history vector.

9. The method of claim 7 wherein said first fetch cycle precedes the second fetch cycle by three fetch cycles.

10. The method of claim 7 wherein said steps of indexing comprises the step of gating the vector with selected bits of a current instruction address.

11. The method of claim 10 wherein said steps of gating comprise the steps of performing XOR operations.

12. The method of claim 8 wherein said substeps of appending comprise the substeps of shifting a bit into a shift register storing the second vector.

13. Branch processing circuitry comprising:

a shift register for storing a global history vector;

control circuitry for selectively updating a first global history vector stored in said shift register operable to:

determine if a selected group of instructions contains a branch instruction;

maintain said first global history vector in said shift register when the selected group does not contain a branch instruction;

shift a first value into the shift register to generate a second vector if the selected group contains a branch instruction and the branch instruction is predicted as a branch taken; and

shifting a second value into the shift register to generate a second vector when the selected group contains a branch instruction and does not contain a branch instruction predicted as a branch taken.

14. The branch processing circuitry of claim 13 and further comprising a branch history table and circuitry for generating an index to an entry in said branch history table using selected bits from a current address and selected bits of said first vector to retrieve a prediction value stored therein.

15. The branch processing circuitry of claim 14 and further comprising circuitry for updating said second vector when said prediction value results in a misprediction comprising:

a queue for storing said first and said second vectors;

circuitry for accessing said vectors from said queue;

circuitry for indexing said branch history table with said first vector and updating a corresponding entry with a corrected prediction value; and

circuitry for updating a vector in said shift register with said second vector; and

circuitry for shifting the corrected prediction value into said shift register.

16. The branch processing circuitry of claim 13 wherein said branch processing circuitry forms a portion of a single chip microprocessor.

17. A processing system comprising:

a microprocessor comprising:

a branch history table for storing branch prediction values;

a global history shift register for storing a global branch history vector;

logic for generating an index to said branch history table and accessing prediction values stored therein using selected bits of a said branch history vector stored in said shift register; and

control circuitry for updating a said global branch history vector stored in said shift register and operable to:

retain a current vector stored in said shift register when a selected fetch group does not contain at least one branch instruction;

shift a bit of a first value into said shift register to generate an updated vector when the selected fetch group has at least one branch instruction predicted to be a branch taken; and

shift a bit of a second value into said shift register when said selected fetch group contains at least one branch instruction and contains no branch instructions predicted to be a branch taken.

18. The processing system of claim 17 wherein said microprocessor further comprises:

a branch instruction queue having a plurality of entries each associated with a fetch group for storing at least first and second corresponding global history vectors;

circuitry for detecting a misprediction associated with a prediction value retrieved from said branch history table and corresponding to said first global history vector in said branch instruction queue;

circuitry for retrieving said first vector from said branch instruction queue and accessing a corresponding entry in said branch history table to correct said prediction value stored therein; and

circuitry for retrieving and modifying said second vector to generate a corrected vector in said shift register.

19. The processing system of claim 17 wherein said processing system further includes a system memory coupled to said microprocessor by a bus.

20. The processing system of claim 17 wherein said fetch group comprises eight instructions.